



UNIVERSITI PUTRA MALAYSIA

**INTEGRATED ENVIRONMENT FOR SOFTWARE
DOCUMENTATION**

MOHAMED AHMED SULLABI

FSKTM 2000 3

**INTEGRATED ENVIRONMENT FOR SOFTWARE
DOCUMENTATION**

By

MOHAMED AHMED SULLABI

**Thesis Submitted in Fulfilment of the Requirements for
the Degree of Master of Science in the Faculty of
Computer Science and Information Technology
Universiti Putra Malaysia**

July 2000



To everybody who helped and encouraged me to finish this work

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in
fulfilment of the requirements for the degree of Master of Science.

INTEGRATED ENVIRONMENT FOR SOFTWARE DOCUMENTATION

By

MOHAMED AHMED SULLABI

July 2000

Chairman: Abdul Azim Abd. Ghani, Ph. D.

Faculty: Computer Science and Information Technology

Software documentation refers to the information on the various phases of the software. It includes design specification, performance specification, functional specification, source code information, development information, etc. The source code documentation represents the collection of documents that explains, describes the functions, structures, inputs, outputs, etc., and defines the purposes and uses of a particular software program.

Good documentation is the major difficulty of creating a good software and the software project cannot succeed without documentation. Unfortunately, it is quite often no technical documentation is produced. In addition, when documentation is produced, it is often poorly or incompletely written, and may not be kept current. Those factors contribute to the reasons for software failures, to the difficulty of maintaining the software at a later time, or to the high overhead into subsequent product development.

The objective of the research is to provide software developers with a useful practical environment for their performance improvement. This environment is an integrated environment that concentrates on solving some of the existing problems, which discourage software developers to document their work; mainly, documentation costs time due to the separation between the software development area and software documentation area, when the programmers should document, and what they should document.

The integrated environment will firstly, provide an encouragement environment for software developers to document their work by combining the development and documentation environments into one environment, and this combination will ease the movement between the two environments in order to reduce the time needed. Secondly, it will integrate the facilities needed to manage the software project and to help the developers determine when documentation should be written and what should be written. The integrated environment has been implemented in a tool called IESD (Integrated Environment for Software Documentation).

The tool was evaluated by a group of postgraduate students to test the workability, usability, and reliability of the system, and verify whether the system had achieved its objectives. Questionnaires were distributed to the students. The analysis of the student responses had shown out that the tool was very useful and easy to use, and the rate of agreement was over 80%.

Based on this study, it can be concluded that the integration between the programming environment and the documentation environment with the facilities provided, has helped the users to tackle the crucial problems of documentation.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia
sebagai memenuhi keperluan untuk ijazah Master Sains.

**PERSEKITARAN BERSEPADU
UNTUK DOKUMENTASI PERISIAN**

Oleh

MOHAMED AHMED SULLABI

Julai 2000

Pengerusi: Abdul Azim Abd. Ghani, Ph. D.

Fakulti: Sains Komputer dan Teknologi Maklumat

Dokumentasi perisian merujuk kepada maklumat pelbagai fasa sesuatu perisian itu. Ia termasuklah spesifikasi reka bentuk, spesifikasi prestasi, spesifikasi fungsian, maklumat kod sumber, maklumat pembangunan, dan lain-lain. Dokumentasi kod sumber mewakili koleksi dokumen yang menerang dan menghuraikan fungsi, struktur, input, output, dan lain-lain serta mendefinisikan tujuan dan kegunaan program perisian berkenaan.

Dokumentasi yang sempurna merupakan masalah utama dalam mencipta perisian yang baik dan tiada projek perisian yang berjaya tanpa dokumentasi. Namun demikian, seringkali dokumentasi teknikal tidak disediakan. Malahan apabila dokumentasi dihasilkan, ia tidak ditulis dengan lengkap dan tidak terkini. Faktor-faktor ini telah menyumbang kepada kegagalan sesuatu perisian hingga sukar untuk mengendalikan perisian tersebut pada masa hadapan, atau mengarah kepada perbelanjaan yang tinggi untuk pembangunan produk seterusnya.

Penyelidikan ini dilaksanakan bagi tujuan menyediakan satu persekitaran praktikal yang berguna kepada pembangun perisian untuk membaiki prestasi mereka. Persekitaran ini merupakan persekitaran bersepadu yang menjurus kepada penyelesaian masalah sedia ada yang melemahkan pembangun perisian mendokumentasikan kerja-kerja mereka; terutama daripada segi nilai masa untuk pendokumentasian kerana pemisahan di antara bidang pembangunan perisian dan bidang pendokumentasian perisian, iaitu bila dan apa yang patut didokumenkan oleh pengaturcara.

Persekitaran Bersepadu akan pertama sekali, menyediakan persekitaran yang menggalakkan untuk pembangun perisian mendokumenkan kerja-kerja mereka melalui penggabungan persekitaran pembangunan dan persekitaran pendokumentasian dalam satu persekitaran, dan gabungan ini akan memudahkan perpindahan di antara dua persekitaran dan ini akan dapat menjimatkan masa yang diperlukan. Kedua, menyepadukan kemudahan-kemudahan yang diperlukan untuk mengurus projek perisian dan membantu pembangun perisian menentukan bila dokumen patut ditulis dan apa yang patut ditulis. Persekitaran bersepadu tersebut telah diimplementasikan pada sebuah peralatan yang diberi nama IESD (persekitaran bersepadu bagi dokumentasi perisian).

Peralatan tersebut telah diuji oleh sejumlah pelajar ijazah lanjutan untuk menguji kebolehan kerja, kebolehgunaan, dan kebolehpercayaan daripada sistem itu dan menentusahkan sama ada sistem telah mencapai matlamatnya. Soal selidik dibahagikan kepada pelajar-pelajar. Analisis respon pelajar telah memperlihatkan

bahawa peralatan tersebut sangat berguna dan mudah untuk digunakan, dengan tingkat persetujuan lebih dari pada 80 %.

Berdasarkan pengajian ini, dapat di buat kesimpulan bahawa persepaduan di antara persekitaran pengaturcaraan dan dokumentasi dengan fasiliti-fasiliti yang disediakan telah membantu pengguna bagi menangani masalah-masalah penting untuk dokumentasi.

ACKNOWLEDGEMENTS

In the name of Allah the Most Beneficent and the Most Merciful. Alhamdulillah and thanks to Almighty Allah for giving me the opportunity and the ability to continue my study, and the patience and perseverance to complete this research.

Many discussions, support and contribution have accompanied this work. I am indebted to many people whom without their comments and support, I wouldn't be able to finish this work. I would like to thank everybody who have helped me to achieve this work.

I would like to thank my supervisor Dr. Abdul Azim Abd. Ghani for his commitment, his invaluable guidance, his understanding, help, and encouragement. Thank you for everything and I would like to express my high appreciation for spending his time that I needed to finish this work.

I would also like to thank Associate Professor Hj. Mohd. Hasan Selamat and Dr. Hajjah Fatimah Ahmad, for their assistances and discussions throughout the research period, which made me improve several aspects of this thesis.

I must thank all members of academic and non-academic staff of the Faculty of Computer Science and Information Technology, and the Graduate School,

University Putra Malaysia for their help and cooperation. A special thanks to some of my friends from the Faculty of Computer Science and Information Technology, who have validated my tool.

I would like to thank my sponsor, the Libyan Ministry of Education, for the financial support throughout the period of my study. The contribution of the staff of the Libyan People's Bureau, Kuala Lumpur, Malaysia, is greatly appreciated. I would like to thank Mr. Mohammed Esmail, the head of Higher Polytech. Inst., Misurata, Libya, and all the academic and non-academic staff of the Institute, who elected me and helped me to continue my study.

Finally, I would like to thank my family members and my friends in Libya for their encouragement and support during my study with special thanks go to my father Ahmed and my mother Fatimah whose faith, love, and 'Doa' have provided me with the foundation to be successful in my study and in my whole life.

Last but not least, throughout the research period, I am accompanied by many friends. Their presence have always made me feel as I am at home. There is no space to mention their names here but I would like to thank all of them and wish them all the best and good luck in their lives.

I certify that an Examination Committee met on 26th July 2000 to conduct the final examination of Mohamed Ahmed Sullabi on his Master thesis entitled “Integrated Environment for Software Documentation” in accordance with Universiti Pertanian Malaysia (Higher Degree) Act 1980 and Universiti Pertanian Malaysia (Higher Degree) Regulations 1981. The committee recommends that the candidate be awarded the relevant degree. Members of the Examination Committee are as follows:

HAMIDAH IBRAHIM, Ph.D.
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

ABDUL AZIM ABD. GHANI, Ph.D.
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

HJ. MOHD HASAN BIN SELAMAT, Ph.D
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

HAH. FATIMAH BT. AHMAD, Ph.D.
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)



MOHD. GHAZALI MOHAYIDIN. Ph.D.
Professor/Deputy Dean of Graduate School
Universiti Putra Malaysia

Date: **22 AUG 2000**

This thesis was submitted to the Senate of Universiti Putra Malaysia and was accepted as fulfilment of the requirements for the degree of Master Science.

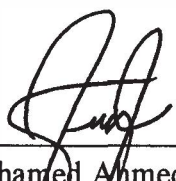


KAMIS AWANG. Ph.D.
Associate Professor/ Dean of Graduate School
Universiti Putra Malaysia

Date: **11 NOV 2000**

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations, which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UPM or other institutions.



Name: Mohamed Ahmed Sullabi
Date: 18-8-2000

TABLE OF CONTENTS

	Page
DEDICATION	2
ABSTRACT	3
ABSTRAK	6
ACKNOWLEDGEMENTS	9
APPROVAL SHEETS	11
DECLARATION	13
LIST OF TABLES	17
LIST OF FIGURES	19
CHAPTER	
1 INTRODUCTION	21
1.1 Background	21
1.2 Project Management Concept.....	22
1.3 Software Project Problems.....	23
1.4 Software Documentation.....	25
1.4.1 Documentation for Software Quality Assurance.....	27
1.4.2 Documentation for Software Maintenance.....	28
1.4.3 Source Code Documentation.....	30
1.5 Research Objectives	32
1.6 Research Methodology.....	33
1.7 Summary of Chapters.....	34
2 LITERATURE REVIEW	36
2.1 Introduction.....	36
2.1.1 Documentation in Project Life-Cycle.....	36
2.2 Role of Documentation.....	37
2.2.1 Problems in the Documentation Process.....	38
2.2.2 Documentation Requirements and Creation.....	41
2.2.3 Document Types and Principles.....	44

2.3	On-Line Software Documentation Tools	47
2.3.1	Documentation Software Tools.....	47
2.3.2	Re-documentation Software Tools.....	54
2.4	Identifying Main Features.....	59
2.5	Research Framework.....	63
2.5.1	Implementing a Documentation Software Tool.....	63
2.5.2	Evaluating IESD.....	64
2.6	Summary.....	65
3	THE INTEGRATED ENVIRONMENT FOR SOFTWARE DOCUMENTATION (IESD) DESIGN AND STRUCTURE.....	66
3.1	Introduction.....	66
3.2	IESD Requirements.....	66
3.2.1	Software Requirements.....	67
3.2.2	Hardware Requirements.....	67
3.3	The IESD Application.....	68
3.4	The IESD Structure.....	69
3.5	The Database.....	70
3.6	Defect Types.....	71
3.7	Measurement of Written Material.....	72
3.8	The IESD Architecture.....	73
3.9	Documentation Guidelines.....	76
3.10	Interacting with the Application (User Interface).....	78
3.11	IESD User Interface Design.....	78
4	IESD EVALUATION AND RESULTS.....	96
4.1	Introduction.....	96
4.2	Usability, Validity and Utility Evaluation.....	96
4.3	Evaluation Method.....	98
4.4	Laboratory Test Study.....	99
4.4.1	Participants.....	100
4.4.2	Environment.....	100
4.4.3	Qualitative Measurements.....	101
4.5	Laboratory Test Results.....	102
4.5.1	The Analysis of Usability Defects.....	103

4.5.2	Learnability and Efficiency of Use.....	103
4.5.3	Subjective Satisfaction Assessment.....	104
4.6	Comparison of IESD with Other On-Line Documentation Tool..	106
4.7	Conclusion.....	109
5	CASE STUDY.....	110
5.1	Introduction.....	110
5.2	The Purpose.....	110
5.3	The Project Registration.....	111
5.3.1	Case 1 Opening.....	111
5.3.2	Case 2 Opening.....	117
5.3.3	Case 3 Opening.....	119
5.4	Documentation Report.....	121
5.5	Conclusion.....	122
6	CONCLUSION AND FUTURE WORK.....	123
6.1	Conclusion.....	123
6.2	Future Work.....	125
	REFERENCES.....	127
	APPENDICES	
A	Questionnaire.....	132
B	Users Response to Questionnaire.....	136
C	Users Response Analysis.....	138
D	Source Code of the Case Study.....	142
E	Documentation Report of the Case Study.....	159
	BIODATA OF THE AUTHOR.....	166

LIST OF TABLES

Table	Page
2.1 Summary of Features of On-Line Software Documentation.....	62
3.1 Required Files for the IESD Application.....	69
3.2 Defect Type Standard.....	71
3.3 Grade Levels of Technical Material.....	72
4.1 Results of the Subjective Satisfaction Assessment.....	105
4.2 Comparison of IESD with Other On-Line Software Documentation Tools.....	108
B.1 Using of Computer.....	136
B.2 Software Documentation Knowledge.....	136
B.3 Users' Answers to Questionnaire.....	136
C.1 Over All Performances.....	138
C.2 Ease of Use.....	138
C.3 Correctness of Operations.....	139
C.4 Usefulness of IESD Application.....	139
C.5 Screen Design & Layout.....	139
C.6 Learning the Documentation Process.....	139
C.7 User Control.....	140
C.8 Consistency.....	140
C.9 Learnability.....	140

C.10	Terminology.....	140
C.11	Operation & Efficiency of Use.....	141
C.12	Users Response Analysis.....	141

LIST OF FIGURES

Figure		Page
2.1	Documentation Passes from Task to Task.....	37
2.2	Document System Architecture.....	51
2.3	InterSect Structure.....	53
2.4	CC-RIDER Visualiser.....	56
2.5	The Architecture of SAMS.....	57
2.6	Example of C Function.....	58
2.7	Display of C Function Using SAMS.....	59
3.1	Structure of IESD Application.....	70
3.2	IESD Architecture.....	74
3.3	Breaking up the Project.....	74
3.4	Timing of the Task.....	75
3.5	The Application Reporting System.....	79
3.6	IESD FlowChart.....	80
3.7	Entering a New Project Flowchart.....	81
3.8	Opening a Project Flowchart.....	83
3.9	Scheduling Flowchart.....	84
3.10	Documentation Flowchart.....	86
3.11	Technical Documentation Flowchart.....	86
3.12	Documentation of Coding Problems Flowchart.....	87
3.13	Reporting Flowchart.....	88

3.14	The Technical Report.....	89
3.15	The Coding Problem Report.....	89
3.16	The Technical and Problem Report.....	90
3.17a	The Technical Report Format.....	92
3.17b	The Coding Problem Report Format.....	93
3.17c	The Technical and Problem Report Format.....	94
4.1	Five-Points Likert Scale Ratings.....	101
5.1	The IESD Registration Form.....	112
5.2	Scheduling Message.....	112
5.3	The IESD Scheduling Form.....	113
5.4	Documentation Message for Case 1.....	114
5.5	Documentation Form with Case 1.....	115
5.6	Switching Between Documentation and Programming via Taskbar.....	115
5.7	IESD Main Menu.....	116
5.8	Documentation Types.....	116
5.9	Documentation Form with All the Project Tasks.....	117
5.10	Documentation Message for Case 2.....	118
5.11	Documentation Form with Case 2.....	119
5.12	Documentation Message for Case 3.....	120
5.13	Documentation Form with Case 3.....	120
5.14	Report Types.....	121
5.15	The Technical Report Dialog Box.....	122

CHAPTER 1

INTRODUCTION

1.1 Background

No management system is of value unless it provides a solution to the problem to which it is directed. But more than that, it must also provide a solution which takes into account the role and function of management itself. The most sophisticated computer solution to a problem is of little worth if it cannot be effectively used, implemented, and controlled by management. Management means many things to many people, making profit in a corporation, winning, and so on. These undertakings are very different, and yet the function is generalised enough to encompass each one of them.

Essentially, the function of management can be defined as:

1. Selecting the objectives of the project.
2. Determining the requirements to meet these objectives.
3. Judiciously allocate the available resources to achieve the objectives according to a plan and schedule.

4. Controlling the entire process from the point of decision or commitment to the point of completion (achievement of objectives).

The function of management is best performed with a proper balance between subjective ability and objective method, its effectiveness is measured by the results achieved and, more especially, by the response time of manager and method when things go wrong.

The purpose of management techniques is not to encroach on the management function, but to provide the tools necessary for it to perform effectively. However, besides establishing broad generalised plans, there is the equally vital and more specific task of planning, scheduling, and supervising the various individual projects, which are integral parts of the overall plan. Efficient planning of these projects is the difference between “on-time” and “late”, and it can mean the difference between success and failure.

1.2 Project Management Concept

Many projects go by considering that the phrase “Project management”, in this context, the success of a system development project will often depend on the duration of the project, the amount of excess resources available, how powerful the user is? Rarely does success have much to do with the quality of the projects

produced or the final delivery schedule and costs (King, 1992). Although it is agreed that we cannot control the software process unless we can measure it, there is some disagreement as to precisely what should be measured (Fenton, 1994).

King (1992) said:

“I believe it’s because many of these projects are not managed properly or at all. It is often been said that we can only manage things that can be measured. Therefore, if we cannot measure what we create, subjective and indirect factors often determine the success or failure of the endeavour.

Accordingly, to manage these activities effectively, we need to set up an environment where we can accurately measure and constantly monitor the efforts against a predetermined set of standards and values. Then we can manage! Certainly, not all software development failures could have been avoided by attention to correct project management, but it surely could have helped.”

1.3 Software Project Problems

Many system development projects fail, as measured by one criterion or another. King (1992) defined a project as having failed if it fails to meet the user’s minimum requirements, or implemented too late to be effective. There are sometimes

purely problems or reasons for project failures, and these are the most unpredictable and least preventable. Nevertheless, for these and other reasons, software projects do fail.

These are some of the problems, which the projects may be suffering from:

1. Projects run late and they cost more than was originally expected (Horberg, 1994).
2. A project may be found to go out of control, due to size of the project.
3. Software development environments and document development environments have remained quite separate (Walker, 1988; Galt & Jones, 1993).
4. Poor documentation (King, 1992).
5. Writing documentation often at the end of the project (Brown, 1989).

Such problems are not inevitable. A well-structured formal approach to the management of project, irrespective of their size, will allow monitoring of progress and costs against the plan and will give early feedback. This can allow a suitable action to be taken to minimise the effect. The result is a project, which is more likely to run according to schedule and meets its budget.